

EXPÉRIMENTATIONS VIRTUELLES: VIE ARTIFICIELLE POUR LA GÉNÉRATION DE FORMES ET DE COMPORTEMENT

Hervé Luga

Equipe VORTEX -IRIT UMR 5505

Université de Toulouse

UT1, 2 rue du doyen Gabriel Marty 31042 Toulouse, France

herve.luga@univ-tlse1.fr

RÉSUMÉ

Cet article présente une sélection de travaux réalisés par l'équipe VORTEX de l'IRIT depuis 1993 dans les domaines de la génération automatique de formes et de comportements. Nous montrons ici la spécificité de ces travaux utilisant des techniques originales issues de la vie artificielle afin de proposer un nouveau type d'interaction entre l'utilisateur et l'environnement de simulation.

1.INTRODUCTION

L'équipe VORTEX de l'IRIT réalise des recherches depuis une vingtaine d'années dans les domaines de la modélisation et dans celui de la création de comportements pour acteurs autonomes en espace virtuel. Les premiers travaux de cette équipe utilisaient des approches traditionnelles de conception de systèmes de type procédurales ou déclaratives. L'approche déclarative propose ainsi de créer des objets ou des scènes tridimensionnelles par description des propriétés de l'objet et de ses liens avec le reste de l'environnement. La génération de comportements autonomes pour les acteurs virtuels¹ proposait une approche de type intelligence artificielle basée sur un système de règles PROLOG.

L'ensemble de ces travaux fournissent des résultats utilisables mais obligent le créateur du système à un travail important de spécification et d'analyse du système généré et de ses interactions possibles avec son environnement. Ce travail, particulier à chaque système généré ne nous semblait pas opportun car il s'est vite révélé peu robuste à des changements environnementaux et donc incapable d'adaptation. Nous avons donc décidé de nous tourner vers des algorithmes issus d'un champ de recherche (alors) émergent appelé « vie artificielle ». Cet article présente une sélection de travaux basés sur ce type d'approche et montre comment cette dernière nous a permis d'ouvrir de nouvelles perspectives dans l'interaction entre interacteurs réels, acteurs virtuels,

avatars et environnement dans le domaine de l'animation comportementale et de la modélisation.

Dans une première partie nous présenterons les principaux algorithmes de vie artificielle en proposant une catégorisation de ces derniers. Nous montrerons ensuite des exemples de travaux dans le domaine de la modélisation et dans le domaine de la génération de comportements autonomes. Avant de conclure nous montrerons comment ces travaux nous ont mené vers la production automatique d'entités virtuelles complètes.

2.LA VIE ARTIFICIELLE

Nous pouvons identifier plusieurs types de systèmes aptes à améliorer les systèmes de production en environnement virtuel. Certains vont se focaliser sur l'aspect interface en essayant de proposer un système le plus efficace possible pour l'interaction entre le créateur et le système. D'autres vont utiliser des systèmes issus de l'intelligence artificielle classique² afin d'aider le créateur à produire, selon un ensemble de contraintes, les objets ou les comportements qu'il souhaite assigner aux acteurs virtuels. Une autre approche consiste à placer le système en situation en le dotant de mécanismes à réaliser son adaptation à son milieu. Ce type de mécanismes est regroupé sous le champ de recherche identifié comme « la vie artificielle ».

Le terme « vie artificielle » a été proposé par Chris Langton (24) qui a identifié cette thématique de recherche comme celle de l'étude des systèmes émergents. On identifie l'émergence lorsque le comportement global d'un système n'est pas « simplement » déductible du comportement local de ses constituants. On va trouver dans cette approche un ensemble de techniques ayant en commun une inspiration « naturelle ». Nous entendons par ce terme que la plupart de ces techniques sont des simplifications ou des imitations de fonctionnement de systèmes réels, le plus souvent biologiques, avec en filigrane l'idée que la vie a fourni aux êtres vivants des solutions pour leur permettre de s'adapter à notre environnement terrestre et

¹Nous nous plaçons alors dans un type d'animation nommé « simulation comportementale » où des acteurs virtuels autonomes agissent par interaction située dans leur environnement.

²Par exemple des systèmes experts ou des réseaux d'automates comme dans le système de simulation comportementale HPTS (25).

que nous allons pouvoir transposer ces caractéristiques à des systèmes artificiels.

2.1. Dichotomie des systèmes de vie artificielle

La vie artificielle est matérialisée par un ensemble de paradigmes qui se traduisent sous la forme de techniques informatiques. Ces dernières sont souvent obtenues par une analogie réalisée entre des systèmes existants et une adaptation de ces systèmes sous forme d'algorithmes. L'inspiration vient souvent de processus biologiques comme nous le verrons par exemple dans le cas des réseaux de neurones mais peut aussi être trouvée dans des systèmes chimiques ou bien encore dans des systèmes techniques mis au point par des êtres humains.

Il est important de noter que de simplifications en optimisations diverses, les méthodes réellement utilisées n'ont fréquemment plus de rapport avec les systèmes qui les ont inspirées. Dès lors comment classer ces paradigmes de vie artificielle ? En étudier l'origine pour les rapprocher n'est pas opportun car un même système réel peut avoir donné lieu à des techniques très différentes. Nous proposons donc ici d'utiliser une approche fonctionnelle en discriminant les systèmes en fonction de leur fonctionnement vis à vis des structures sur lesquels ils portent.

2.1.1. Des systèmes générateurs

Un système générateur est un système qui va agir de manière incrémentale sur une structure de données.

L'initialisation de l'algorithme est réalisée par la création d'un état initial qui est souvent désigné sous le terme d'axiome. L'évolution du système est obtenue par application successive de règles. A un état courant de la simulation est ensuite associé un élément de l'espace des états. Le produit d'un système générateur après n pas de simulation est donc la composition de l'action des règles sur cet axiome. Cette définition implique une fermeture de l'espace E par la fonction induite par l'application récursive de l'algorithme.

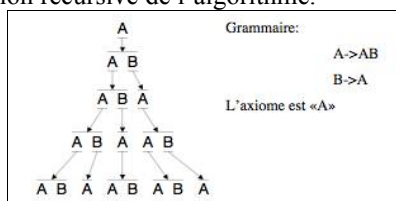


Figure 1. Exemple d'application itérative d'un système générateur.

En rapprochant cette définition des systèmes vivants on peut clairement classer les systèmes générateurs comme des systèmes agissant au niveau endogénétique. Il n'existe en effet pas de modification endogène des règles ni de l'axiome, seul l'état courant est modifié. De plus, si le système n'est pas dépendant du hasard ou d'actions extérieures il est parfaitement déterministe. Les systèmes générateurs les plus anciens sont les

modèles de réaction diffusion (39) et les automates cellulaires de Ulam et Von Neumann. Ils sont une simplification de l'évolution des concentrations de composés chimiques en solution. Intégrés par la définition généralisée des IFS¹ qui permet de les lier aux fractales, ces modèles sont aujourd'hui encore les supports de nombreuses études sur la complexité. Nous pouvons classer dans ces modèles les systèmes de masse ressort ou encore les systèmes à base de grammaires tels que les L-Systèmes dont nous allons développer une application dans l'exemple suivant.

2.1.2. Des systèmes d'évolution

Un système d'évolution va porter sur une population d'individus codés par une structure de données dont l'expression est la solution potentielle à un problème. L'évolution va modifier cette population en manipulant cette structure de manière à rendre son expression globalement plus adaptée à son environnement.

Ce faisant, les meilleurs individus de cette population convergeront peu à peu vers un maximum d'adaptation qui sera une solution, parfaite ou approchée, au problème posé. Pour cela les systèmes d'évolution utilisent en général une simplification du processus d'évolution naturelle proposé par Charles Darwin.

Un système d'évolution type comporte au moins trois opérations itérées jusqu'à convergence de l'algorithme:

- Une opération de notation de la population: Cette opération consiste à attribuer à chacun des individus de la population une note représentant son adaptation face au problème considéré. Cette note d'adéquation est souvent référencée sous le terme de «fitness» des individus de la population. C'est l'attribution de cette note qui permet de poser le problème vers la solution duquel l'algorithme doit converger.

- Une opération de sélection: Cette opération va permettre de sélectionner les individus dont les caractéristiques vont pouvoir se propager dans la population. Elle va permettre de régler la pression de sélection et ainsi de définir la vitesse de convergence.

- Des opérateurs de modification permettant de modifier les individus de la population de manière à créer une nouvelle population à évaluer. Les opérateurs classiquement utilisés peuvent être classés en deux groupes: des opérateurs dits de croisement permettant de générer de nouveaux individus à partir d'au moins deux parents et des opérateurs de mutation agissant par modification de l'individu sélectionné. Ces opérateurs ne sont pas forcément utilisés dans tous les systèmes d'évolution, certains, comme les stratégies d'évolution, se contentant d'un processus itératif de mutations.

¹Iterated Function Systems: systèmes incrémentaux basés sur la répétition d'une même fonction sur un environnement.

- A ces opérateurs itératifs il convient d'ajouter un opérateur de création de la population initiale permettant de générer de nouveaux individus. Cet opérateur peut fournir un individu de manière aléatoire parmi l'espace d'états que permet de représenter le code du génotype des individus ou ne permettre de ne générer qu'un sous ensemble de cet espace.

Ces opérateurs portent sur deux niveaux complémentaires de représentation des individus. Un individu est ainsi défini par son génotype sur lequel vont porter les opérateurs de modification. Une fonction de représentation transforme ce génotype en un phénotype qui va être testé face au problème considéré et donc fournir la note d'adaptation. La fonction d'évaluation qui va définir le chemin de convergence de l'algorithme peut donc être vue comme la composée entre la fonction transformant le génotype en phénotype par la fonction associant une note d'adaptation à ce phénotype. Cette définition explique l'importance du codage sur la facilité de convergence de l'algorithme. Ce codage devra donc faire l'objet d'un choix particulier lors de son choix afin de favoriser l'apparition de blocs constitutifs à fort potentiel qui seront les briques de base assemblées pour produire les individus solution au problème. De même, les opérateurs de modification sont en général adaptés au problème considéré afin d'accélérer le processus de convergence.

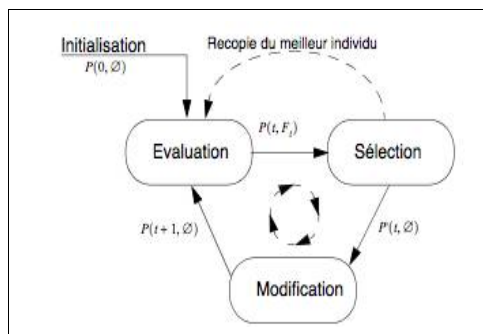


Figure 2. Schéma synoptique de fonctionnement d'un système évolutionniste.

L'itération a lieu jusqu'à «convergence» de l'algorithme. P_t est la population à l'itération t , F la note d'adaptation des individus.

Un des problèmes majeurs des systèmes d'évolution est le choix de la fonction permettant définir la fin de l'algorithme. En effet, si le problème considéré ne comporte pas de solution connue, l'algorithme se contentera souvent d'une solution approchée dont l'exactitude est laissée au créateur de l'algorithme. Il existe ainsi de multiples stratégies d'arrêt de ces algorithmes parmi lesquelles nous pouvons citer:

- L'arrêt de la convergence au bout d'un nombre fixé de générations.

- L'arrêt lorsque l'évaluation d'un individu atteint une note fixée.

- L'arrêt lorsque la note du meilleur individu ne progresse plus au delà d'un certain nombre d'évaluations.

- L'arrêt lorsqu'un temps ou un nombre fixé d'évaluations a été atteint.

Le plus souvent il sera obligatoire de lancer plusieurs simulations permettant d'explorer différentes voies d'évolution afin d'approcher une convergence de l'algorithme vers un maximum global. Un seul lancement de l'algorithme ne permettant en général pas de valider que la solution proposée correspond à un tel critère.

Les systèmes évolutionnistes les plus connus sont sans contexte les algorithmes génétiques dont la paternité revient à John Holland (16) mais dont la redécouverte en 1989 par David Golberg (14) a ouvert la voie à un large champ applicatif. Il existe néanmoins de multiples variantes de ces algorithmes qui diffèrent au niveau de la taille de population mais aussi au niveau du type de structure de données utilisée comme support de l'information portée par les individus. Parmi ces variantes nous pouvons discerner « Genetic Programming » proposée en 1993 par John Koza (21,22) qui permet de manipuler de manière directe des structures arborescentes permettant de représenter par exemple des fonctions ou des programmes informatiques. Cette dernière technique et ses nombreuses variantes fournit un champ d'utilisation très vaste allant de la création automatique de circuits électroniques à la génération de comportements. Elle excelle aussi dans le domaine de la recherche de solutions compétitives face à celles produites par des êtres humains¹.

2.1.3. Des systèmes d'apprentissage

Un système d'apprentissage va agir sur une structure de données définissant le comportement d'un système en modifiant cette structure de manière à adapter le comportement à un environnement. L'adaptation se fait généralement en fonction d'un «feedback» de la part de cet environnement vers le système d'apprentissage pour moduler la modification de la structure de données.

Si la phase d'apprentissage est réalisée de manière explicite par utilisation de couples entrées/sorties connues l'apprentissage sera dit supervisé. Dans le cas où l'apprentissage se fait de manière autonome par auto adaptation du système à son environnement au cours de son utilisation on parlera d'apprentissage non supervisé.

Le «feedback» peut être exogène au système apprenant et on aura alors un type d'apprentissage par interaction

¹Voir les sessions Human Competitive Results de la conférence GECCO.

directe entre l'environnement et le système. Il peut aussi être endogène et donc interne au système apprenant.

L'exemple le plus connu de système d'apprentissage est sans conteste la famille des réseaux de neurones artificiels. Introduits dans les années 1950 par (15), ces derniers prennent leur inspiration dans une idéalisation du fonctionnement du cerveau. Nous utilisons aussi dans notre équipe un autre type de systèmes d'apprentissage nommé « systèmes de classeurs » qui, proposés par John Holland en 1975 ont réellement été utilisés depuis les travaux de Wilson en 1994 (43). Ces systèmes permettent de faire évoluer automatiquement une base de règles en fonction de contraintes matérialisées par le « feedback » de l'environnement.

2.2.L'émergence dans les systèmes de vie artificielle

Une des caractéristiques principales des systèmes de vie artificielle est l'interaction qu'ils impliquent entre le système, l'interacteur et les entités simulées. Cette interaction ne sera pas de manière générale de type impératif mais sera plutôt une collaboration entre ces trois parties pour former la solution à un problème.

Ainsi le créateur du système créera le problème à résoudre en spécifiant l'environnement d'évolution et parfois des fonctions d'évaluation. Le système évoluera alors de manière indépendante en fonction de ses propres perceptions et donc en interaction avec les autres entités virtuelles présentes dans l'environnement. On peut remarquer ici une grande intrication entre tous ces éléments. C'est ce couplage fort système évoluant/environnement qui permet aux techniques de vie artificielle de produire des solutions robustes et adaptables en environnement dynamique car elles ne placent pas l'objet évoluant dans un environnement idéalisé et donc forcément biaisé. Le revers de ces remarques est qu'il est parfois difficile d'obtenir un objet correspondant parfaitement à un souhait précis car le créateur aurait alors à imaginer tous les types d'interactions que cet objet pourrait rencontrer lors de sa génération.

3.EXPÉRIMENTATIONS EN ENVIRONNEMENT VIRTUEL

3.1.Elevage de sculptures virtuelles

Ce projet réalise le lien entre les systèmes évolutionnistes et le processus de création artistique. L'objectif est de créer un nouveau type d'outil de création mettant en oeuvre des processus imités des systèmes d'évolution permettant à l'artiste de développer une nouvelle relation avec l'objet créé. Cette problématique se pose pour de nombreux types de créations (scénarii, textures, Musique, Effets vidéo,...) et a déjà été proposée par William Latham et Karl Sims puis explorée par de nombreux créateurs (9,19).

Nous avons décidé dans notre travail de nous focaliser sur la génération de sculptures virtuelles et nous limiterons dans la suite à l'étude des solutions mettant en jeu la modélisation ou les opérateurs que nous avons choisis: les surfaces implicites et les systèmes d'évolution.

3.1.1.Travaux précurseurs afférents à cette application.

Nous pouvons discerner plusieurs types de travaux faisant le lien entre formes tridimensionnelles et opérateurs évolutionnistes. Peter Bentley (4) distingue ainsi quatre catégories d'applications mêlant génération d'objets et vie artificielle:

- Les travaux mettant en oeuvre les systèmes évolutionnistes à des fins d'optimisation de formes. Même si nous avons déjà mené des études dans cette optique (30), l'application que nous développons ici ne s'y inscrit pas.
- Les travaux portant sur la génération de créatures artificielles.
- Les travaux utilisant les techniques de vie artificielle pour l'aide au design.
- Les travaux utilisant ces systèmes dans un processus de création artistique. Le système sert alors à l'artiste à exprimer son art. Cette catégorie est très proche de la précédente si ce n'est qu'elle a une finalité pratique, les créations n'étant pas nécessairement des objets réels.

C'est dans cette dernière, optique, en liaison entre arts et science que nous nous situons. Le bref état de l'art que nous présentons prendra néanmoins en compte les autres aspects car notre travail s'est inspiré de projets appartenant à l'ensemble de ces catégories.

William Latham a proposé dès 1989 (38) un système d'évolution de formes tridimensionnelles basé sur des opérateurs issus de la vie artificielle. Son projet, nommé MUTATOR, permet de générer des formes définies par L-Systèmes sous la direction d'évaluations effectuées par l'utilisateur.

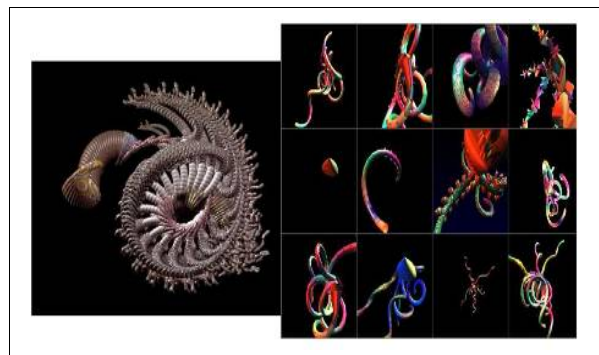


Figure 3. Formes « organic art » de Latham.

Les résultats obtenus sont des formes variées à l'aspect organique accentué par un rendu aux couleurs naturelles dans la plupart de ses créations. Ce projet a permis de

montrer la variété des formes que peuvent générer de tels outils.

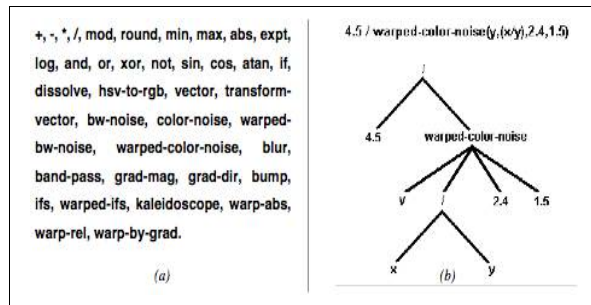


Figure 4. Le vocabulaire (a) et un arbre représentant une texture (b) selon Karl Sims.

En 1991, Karl Sims (33) a proposé l'utilisation d'opérateurs de croisement et de mutation issus de l'évolution artificielle pour la création de textures, structures et animations. L'objectif poursuivi ici était d'offrir un nouveau type d'interaction pour les créateurs en leur permettant de manipuler non plus une seule composition mais d'interagir avec un groupe de compositions.

Suivant les travaux précédents de nombreux auteurs se sont penchés sur une dualité génotype/phénotype pour la génération d'objets, leur objectif étant plus souvent l'optimisation que l'art. Ils ont pour cela proposé d'utiliser des structures de données variées comme support génotypique. Bentley (4) propose ainsi dans son modèleur GADES de représenter une forme comme une agglomération de polyèdres. Le génotype est alors la position et la taille de ces blocs. D'autres auteurs (3) ne codent que les paramètres de surfaces. L'évolution est alors contrainte par la famille de courbes mises en jeu. Parmi ces dernières une catégorie de surfaces nous a particulièrement intéressés car elle permet de générer une grande variété de formes avec une formule compacte: Les surfaces implicites.

3.1.2. Les surfaces implicites

Une surface implicite est une surface définie comme l'iso-potentielle d'une fonction mathématique. Ainsi, si nous considérons une fonction f appelée fonction potentiel et un scalaire V fixé, une forme tridimensionnelle sera définie par le lieu des points de \mathbb{R}^3 tels que $f(x,y,z)=V$. Comme nous pouvons le remarquer la fonction de potentiel définit une valeur en tout point de l'espace ce qui entraînera des effets induits lors du rapprochement de plusieurs potentiels (effets de fusion notamment).

Cette définition mathématique des fonctions implicites est très puissante car elle permet de générer virtuellement n'importe quelle forme d'objet. Elle pose cependant des problèmes pour la création car leur utilisation n'est pas très intuitive et pourrait mener à une création « aléatoire ».

Une autre méthode d'interaction avec ce type de surfaces a été proposée par Brian Wyvill (2). Dans son modèle on compose des surfaces implicites définies à l'aide de squelettes à l'aide d'opérateurs de CSG classiques¹.

3.1.3. Notre modèle d'interaction

En s'inspirant des travaux précédents nous proposons un modèle permettant de tirer parti des possibilités offertes par les deux méthodes de définition des surfaces implicites que nous venons d'évoquer.

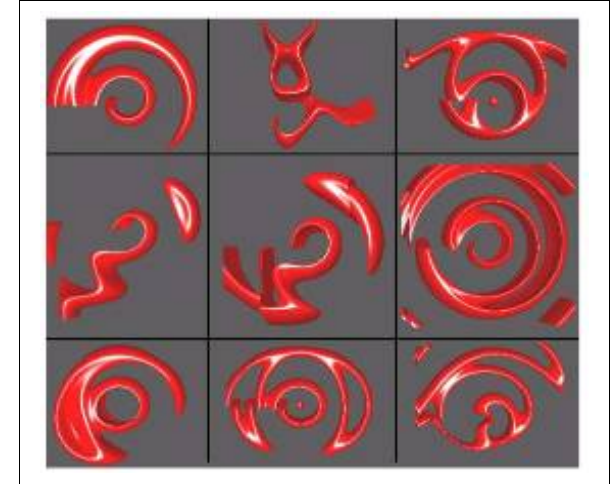


Figure 5. Mutations au niveau mathématique.

Nous proposons donc d'utiliser un système d'interaction à deux niveaux:

- Au niveau algébrique: Des formes de base sont définies à l'aide d'équations mathématiques qui constituent le génotype des formes. Les variables utilisées sont les coordonnées cartésiennes et polaires des points ainsi que des constantes aléatoirement fixées lors de leur création. Les opérateurs utilisés sont des fonctions unaires ($-$, \sin , \cos , 2 , 3 , $^0.5$, ...) ou binaires ($+$, $-$, $*$, $/$, ...) auxquelles nous avons ajouté des opérateurs inspirés des travaux de Karl Sims précédemment cités. Ce dernier choix se justifie par l'adéquation entre ce type d'opérateurs et le processus de création. Nous avons ainsi créé des opérateurs de répétition, de déformation et de récursion. Le support de définition de ces fonctions mathématiques est un arbre. La création des éléments se trouvant à ce niveau peut être réalisée de manière directe par l'entrée d'une équation mais aussi par composition à l'aide d'opérateurs de croisement et de mutation portant sur la population des surfaces mathématiques. Dans notre modèle ce niveau mathématique constituera un ensemble de primitives.
- Au niveau CSG: Les surfaces définies précédemment vont être composées à l'aide d'opérateurs de type CSG. Ici encore c'est une structure d'arbre qui servira de support à ce codage. Ces arbres pourront être

¹Opérateurs de type union, intersection, mélange, extrusion, ...

directement créés à l'aide de l'interface ou encore être produits par croisements et mutations.

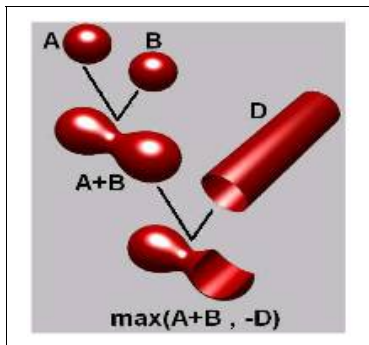


Figure 6. La légende de la figure devra être placée sous la figure.

Afin de permettre un processus de création itératif plus efficace notre système propose une interface basée sur la notion de populations. On trouvera ainsi quatre populations différentes:

- Une population des formes algébriques de base.
- Une population de travail au niveau algébrique.
- Une population au niveau CSG.
- Une population de travail au niveau CSG.

Les populations de travail servent à proposer à l'utilisateur différentes applications des opérateurs lorsque ceux ci peuvent proposer plusieurs résultats. Ils seront notamment utilisés pour les opérateurs de croisement et de mutation. Si une forme est jugée satisfaisante elle sera alors recopiée dans la population principale.

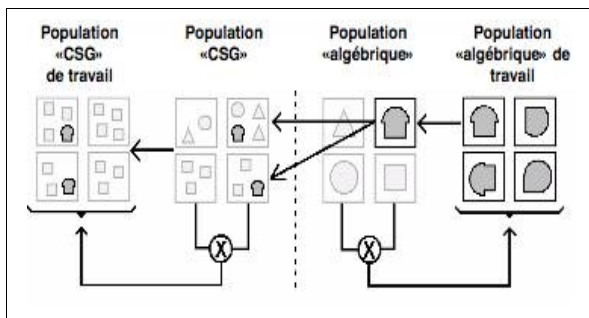


Figure 7. Relation entre les différents niveaux de populations.

Le système maintient de plus le lien entre les différents éléments des populations. Ainsi, toute modification d'une forme au niveau algébrique sera propagée de manière automatique à toute forme du niveau CSG la mettant en jeu.

3.1.4. Critique de cette expérimentation

La définition des surfaces au niveau algébrique apporte une grande richesse dans le processus de création des formes de base mais se révèle vite fastidieux à utiliser si on cherche à réaliser un objectif fixé. Au contraire le niveau CSG est relativement simple à appréhender mais

pose des problèmes de richesse des formes générées lorsqu'on utilise un nombre restreint de primitives. Notre système, en mixant ces deux niveaux, permet à l'artiste d'exploiter les apports de ces deux modèles. Il permet ainsi de générer des formes variées soit à but d'exploration de formes soit avec un objectif guidé. Il permet de plus la création de formes persistantes et évolutives car le génotype est explicite et peut donc être à tout moment retravaillé.

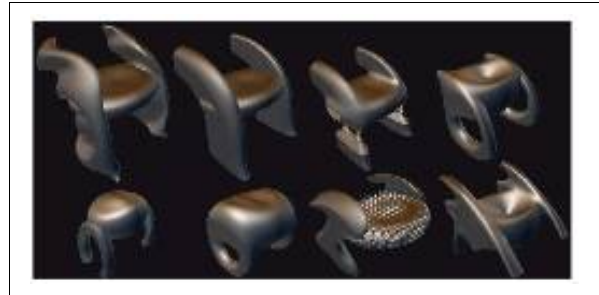


Figure 8. Exemple de génération guidée: population de « chaises ».

Notre travail a donc permis de dégager une nouvelle voie dans les domaines de la création artistique et du design assisté en proposant de nouveaux outils de création. Ces outils, si ils sont compréhensibles n'en sont pas pour autant utilisables par une personne ne connaissant rien aux concepts sous tendant les différents opérateurs et leurs liaisons. Ainsi si l'auteur (Iver Bailly Salins) arrive à exploiter totalement son outil de création il n'en est pas de même de personnes tierces auxquelles nous avons soumis cet outil. Cette remarque pose le problème de la liaison entre un artiste, ses outils et son oeuvre qui sont la plupart du temps totalement intriqués et indissociables.

3.2. Génération de comportements d'acteurs virtuels

Nous allons dans cette partie présenter l'utilisation d'un système d'apprentissage pour l'aide à la génération de comportement d'acteurs virtuels.

3.2.1. Définition du système de simulation

Cette application s'inscrit dans le cadre d'un projet de développement d'un système « classique » de simulation comportementale (37) appliqué aux humains virtuels.

Les caractéristiques d'un tel système sont les suivantes:

- Chaque entité virtuelle possède ses propres capteurs, effecteurs et son propre système comportemental.
- Une entité virtuelle est autonome et n'agit qu'en fonction de ses perceptions.
- Il n'y a pas de différence entre une entité simulée et un avatar.

Le système comportemental est basé sur un ensemble de modules comportementaux. Ces derniers sont organisés de manière hiérarchique et leur enchaînement peut être représenté sous la forme d'un automate. Ces modules fournissent l'ensemble des actions que l'entité virtuelle aura à réaliser et les place dans une liste priorisée qui

sera ensuite exécutée en séquence sauf si une exception oblige à réarranger la liste.

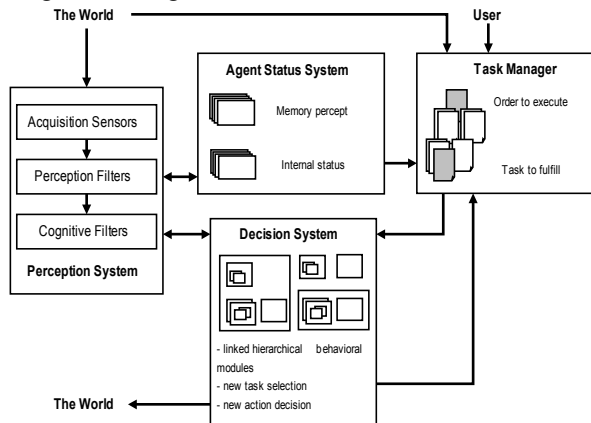


Figure 9. Aperçu de l'architecture du système comportemental.

3.2.2. Un système de classeurs pour la génération de comportements

Notre objectif est donc d'automatiser la génération des modules comportementaux. Nous proposons pour cela d'utiliser des systèmes de classeurs apprenant par renforcement les enchaînements de déclenchement de modules.

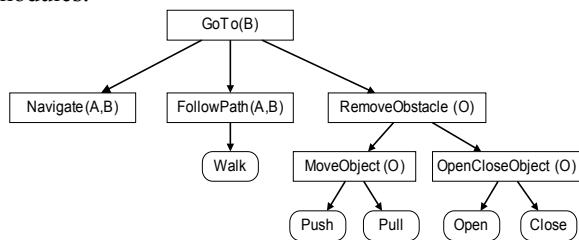


Figure 10. Le module de mouvement: GOTO

Dans une approche ascendante ce système sera utilisé pour créer les liens entre des modules comportementaux pré-existants afin d'en créer un nouveau. Ce module pourra ensuite être utilisé dans une création ultérieure et ainsi accroître la base des modules disponibles. Les règles du système de classeur décrivent les transitions entre les modules en accord avec l'état de l'entité et des entrées issues de l'environnement.

Nous avons dans une première approche appliqué notre système à la génération « offline » de modules. L'apprentissage est réalisé par renforcement. Cela signifie que nous évaluons le comportement du personnage apprenant en situation dans un environnement simulé. L'environnement fournissant à l'apprenant une récompense associée à chaque action réalisée dans l'environnement. Après quelques pas de simulation le système a convergé vers un ensemble de paires condition/action qui peuvent être visualisées comme un automate. Cet automate offre donc une vue synthétique du nouveau comportement produit.

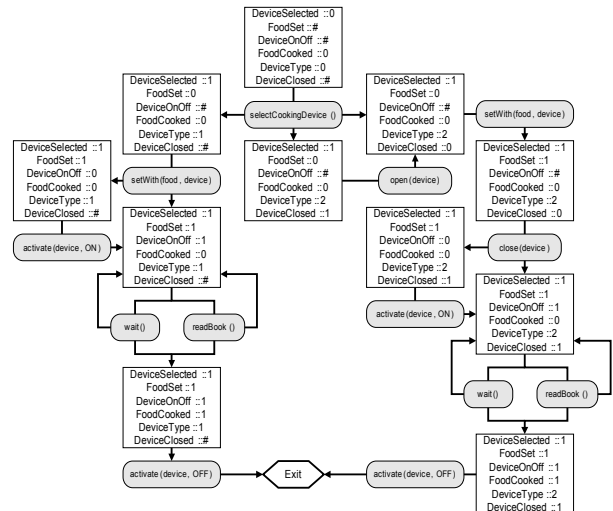


Figure 11. Un automate généré automatiquement: « Cuisiner ».

Après avoir appliqué cette approche à la génération d'un ensemble de modules comportementaux nous avons tenté de produire un automate plus complet en fournissant au système d'apprentissage un ensemble de comportements de base plus conséquent afin de complexifier la phase d'apprentissage. Nous avons aussi testé l'extensibilité de la méthode en ajoutant après apprentissage de nouveaux comportements pouvant être intégrés dans la séquence d'action. C'est par exemple le cas de l'action « readbook » qui a été placée par le système en parallèle de l'action « attendre ».

Ces deux expérimentations ont mis en exergue les possibilités de notre méthode à s'appliquer à des environnements évolutifs.

4. MORPHOGENÈSE DE CRÉATURES ARTIFICIELLES

Les travaux que nous avons développés dans les domaines de la création d'images et de comportements nous ont poussés à étudier la possibilité de génération conjointe de morphologie et de contrôleurs. Nous sommes ainsi intéressés à la génération de créatures artificielles sous contraintes fonctionnelles.

Ces travaux se font dans la lignée de ceux de Karl Sims (34) qui a montré en 1994 que l'évolution artificielle était apte à créer de manière complète des créatures virtuelles¹. De nombreux auteurs dont Maciej Komosinski ou Hod Lipson (20,29) ont depuis prolongé ces travaux.

Dans ces expérimentations les créatures sont plongées dans un environnement 3D simulé comportant non seulement les interactions géométriques mais aussi les interactions physiques² entre les objets de la scène. Le comportement d'une entité est défini en créant des liens entre ses capteurs et ses effecteurs. Ces derniers contrôlent les jointures entre les différentes parties de la

¹Karl Sims travaillait alors pour « Connexion Machines » qui lui fournissait des moyens de calcul parallèle performants.

²Au sens de la mécanique newtonnienne.

créature et permettent ainsi de la faire bouger en modifiant les angles, vitesses ou couples appliqués à ces jointures..

Un système évolutionniste de type algorithme génétique est utilisé pour réaliser l'évolution conjointe de la morphologie et du système de contrôle.

L'originalité de notre travail consiste en une recherche d'interactions plus importantes entre les créatures générées et leur environnement par l'introduction de complexité dans ce dernier.

4.1.1. Définition des créatures artificielles

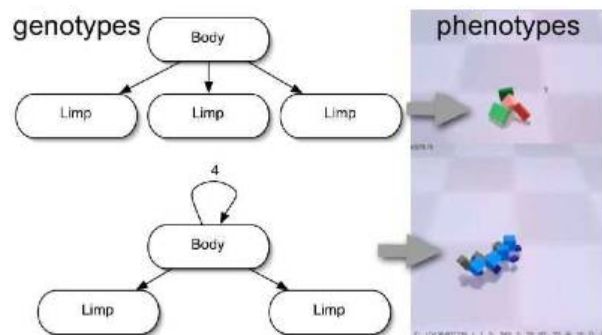


Figure 12. Définition des morphologies sous forme de graphes

La morphologie de nos créatures utilise le modèle original de Karl Sims qui proposait l'utilisation de graphes orientés récurrents qui permettent, à partir d'une forme compacte, de produire une grande variété de morphologies de grande complexité. Nous proposons d'utiliser un nouveau type de contrôleur comportemental basé sur l'utilisation d'un système de classeurs permettant la composition de modèles de signaux servant à contrôler les jointures.

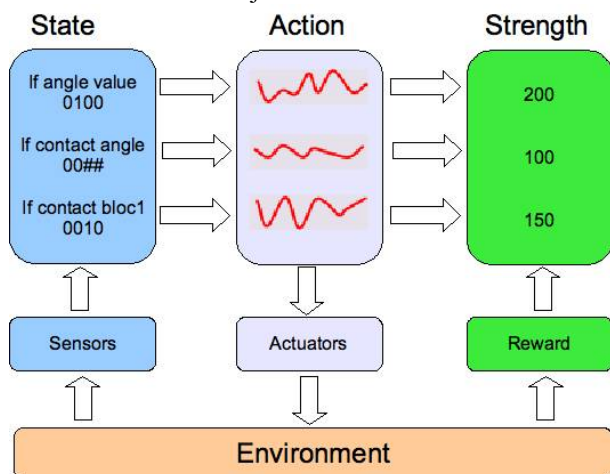


Figure 13. Lien entre capteurs et signaux activés

Chaque expérimentation dure 40 secondes de simulation durant lesquelles une créature est confrontée au problème à résoudre qui constitue sa fonction d'évaluation.

4.1.2. Expérimentations

Nous avons débuté par un problème classique: trouver une créature se déplaçant sur un sol plan. Des résultats encourageants nous ont permis de montrer la validité du modèle proposé.



Figure 14. Exemple de créatures obtenues sur le problème du déplacement.

Nous avons de plus étudié sur cette problématique simple l'effet de la fourniture d'un axe privilégié de déplacement sur la création de morphologies symétriques.

Nous avons ensuite confronté nos créatures à des expérimentations mettant en jeu des objets statiques.

La première expérience menée consistait à faire monter un escalier à nos créatures. Ici encore, la sélection se fait en mesurant la distance parcourue sur l'escalier.

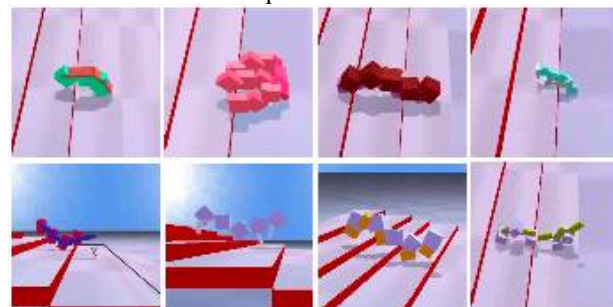


Figure 15. La problématique de la montée d'escaliers.

Nous avons trouvé plusieurs stratégies: certaines créatures développent ainsi des morphologies de taille importante tandis que d'autres privilégient des corps de type « vers » avec beaucoup d'éléments mobiles. Cette application nous a permis de mettre en évidence une autre capacité des algorithmes évolutionnistes: la possibilité de trouver des erreurs dans le programme de simulation. Nous avons en effet mis en évidence une stratégie de déplacement de créature utilisant des erreurs de notre système de gestion de la mécanique newtonienne. Ainsi une lignée de créatures provoquait une division par zéro qui se traduisait par leur éjection par dessus l'escalier et donc leur assurait une note maximale !

Nous avons ensuite étudié l'interaction entre nos créatures et un objet mobile. L'objectif est ici de réaliser le déplacement de la créature à l'aide d'un objet de type « planche à roulettes ».

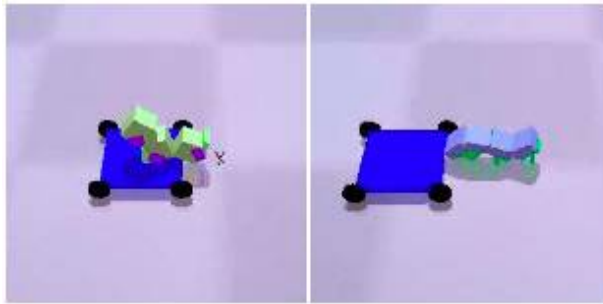


Figure 16. Créatures se déplaçant avec une planche à roulettes.

Cette planche possède 4 roues et l'objectif des créatures est de parcourir la plus grande distance possible en compagnie de la planche. Au démarrage de la simulation la créature est placée sur la planche qui est fixe. Ici encore les lancements successifs de l'algorithme nous ont permis de mettre en évidence une grande diversité de stratégies: certaines créatures lancent la planche et la rattrapent tandis que d'autres montent dessus et la poussent en utilisant une partie de leur morphologie comme propulseur.

4.1.3. Conclusion

Nous avons montré avec ces expérimentations la puissance des systèmes d'évolution et leur capacité à fournir des solutions face à des problèmes de plus en plus complexes. La similarité entre nos créatures qui ont majoritairement une forme de type « vers » est un effet de bord de notre modèle de définition des morphologies et des opérateurs génétiques qui leur sont associés. Nous avons aussi validé le nouveau type de contrôleur basé sur les systèmes de classeurs que nous avons utilisé. Nous travaillons maintenant à la création d'un environnement de type « écosystème » où les créatures en interaction évolueront sous l'effet d'une fonction endogène et non plus d'un but exogène.

5. CONCLUSION

Nous avons présenté dans cet article un ensemble de travaux réalisés dans l'équipe VORTEX de l'IRIT portant sur la génération de formes et de comportements en utilisant des techniques issues de la vie artificielle. En tirant un bilan global de ces expériences nous pouvons dire que le problème majeur soulevé par ces systèmes est l'expérience que nécessite leur mise en oeuvre de manière efficace. La complexité qui existait dans l'étude et la décomposition du système proposée par les systèmes classiques ne disparaît donc pas mais est en partie remplacée par une compétence certaine dans la mise en oeuvre des différents systèmes. On ne parle pas ici simplement de la définition de fonctions d'évaluation mais aussi dans le réglage des algorithmes et dans la définition d'opérateurs cohérents et efficaces dans l'application développée. Il existe néanmoins des cas où ces méthodes sont les seules disponibles et

permettent alors d'obtenir des résultats là où les systèmes classiques sont inopérants ou inefficaces. D'un point de vue de la relation entre le système et ses créateurs ces systèmes sont néanmoins très intéressants car ils proposent réellement de nouveaux types de solutions qui n'ont pas été créés directement par un esprit humain. Reste alors à créer le système de contrôle permettant que la création reste conforme aux besoins réels du créateur.

Nous continuons actuellement les travaux portant sur la génération de créatures en nous plaçant au niveau de la croissance cellulaire. Notre objectif est ici d'ouvrir un nouveau champ de création de systèmes autonomes aptes à s'auto-configurer, s'auto-assembler et s'auto-réparer¹ dans des environnements toujours plus complexes.

6. RÉFÉRENCES

- [1] I. BAILLY SALINS, H. LUGA, Artistic 3D Object Creation Using Artificial Life Paradigms, SmartGraphics International Symposium (SG 2007), Kyoto, Andreas Butz, Brian Fisher, Antonio Krüger, Patrick Olivier (Eds.), Springer, LNCS 4569, p. 135-145, June 2007.
- [2] J. BAJA, J. BLINN, J. BLOOMENTHAL, M.P. CANI, A. ROCKWOOD, A., B. WYVILL, B., G. WYVILL, Introduction to Implicit Surfaces. Morgan Kaufmann Publishers Inc, San Francisco, CA, USA 1997.
- [3] E.J. BEDWELL, D.S. EBERT Artificial evolution of implicit surfaces, ACM SIGGRAPH 98 Conference abstracts and applications, Orlando, Florida, United States, 1998.
- [4] P.J. BENTLEY, Evolutionary Design by Computers, Morgan Kaufmann Publishers Inc, San Francisco, CA, USA (1999)
- [5] B. BLUMBERG, M. DOWNIE, Y. IVANOV, M. BERLIN, P. JOHNSON B. TOMLINSON, Integrated Learning for Interactive Synthetic Characters, SIGGRAPH 02, San Antonio, TX, July 21-26, 2002.
- [6] J. BONGARD, C. PAUL, Investigating morphological symmetry and locomotive efficiency using virtual embodied evolution. In J. - A. M. ET AL. Ed., From Animals to Animats: The Sixth International Conference on the Simulation of Adaptive Behaviour (pp. 16, 93), 2000
- [7] J. BONGARD AND R. PFEIFER, How the Body Shapes the Way We Think: A New View

¹Axe de recherche self*

- of Intelligence. Bradford Books, ISBN-10:0-262-16239-3, 2007.
- [8]J. BONGARD AND H. LIPSON Integrated design, deployment and inference for robot ecologies. In Proceedings of Robosphere, (pp. 19, 20), 2004.
- [9]M. BRET, Virtual living beings, Heudin, J.-C. (ed.) VW 2000. LNCS (LNAI), vol. 1834, pp. 119–134. Springer, Heidelberg, 2000.
- [10]R. BROOKS, Intelligence without representation, Artificial Intelligence 47, 1-3, 139–159. (p. 4), 1991.
- [11]S. CUSSAT-BLANC, H. LUGA, Y. DUTHEN, Using a single cell to create an entire organ, International Conference on Artificial Reality and Telexistence (ICAT 2007), Esbjerg (Denemark), T. Brooks, Y. Ikei, E. Peterson, M. Haller (Eds.), IEEE Computer Society, p. 300-301, november 2007.
- [12]C. DARWIN, On the origins of species by means of natural selection, 1859.
- [13]J. FUNGE, X. TU and D. TERZOPOULOS, Cognitive Modeling: Knowledge, Reasoning and Planning for Intelligent Characters, SIGGRAPH 99, Los Angeles, CA, August 11-13, 1999.
- [14]D.E. GOLDBERG, Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley. (pp. 11, 14, 27), 1989.
- [15]D. HEBB, The Organization of Behavior, 1949.
- [16]J.H. HOLLAND., Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, 1975. Republished by the MIT Press, 1992.
- [17]C. JACOB, H. KWONG, B. WYVILL, Toward the creation of an evolutionary design system for implicit surfaces, Western Computer Graphics Symposium, Skigraph'01, Sun Peaks Resort, British Columbia, 2001.
- [18]M.W. JONES, Direct surface rendering of general and genetically bred implicit surfaces, 17th ann. conf. of eurographics (uk chapter) cambridge, pp. 37–46, 1999.
- [19]Y. KAWAGUCHI, Electronic Art and Animation Catalog, pp. 90–91. ACM Press, New York, NY, USA SIGGRAPH '05: ACM SIGGRAPH, 2005.
- [20]M. KOMOSINSKI, The world of framsticks: Simulation, evolution, interaction. In VW '00: Proceedings of the Second International Conference on Virtual Worlds, (London, UK, 2000), pp. 214–224. Springer-Verlag. (pp. 15, 16, 17, 42, 45, 64, 87), 2000.
- [21]J.R. KOZA, Genetic programming: on the programming of computers by means of natural selection. MIT Press (a Bradford book) Cambridge, MA, USA 1992.
- [22]J.R. KOZA, The genetic programming paradigm: Genetically breeding populations of computer programs to solve problems, Soucek, B., Group, I.R.I.S. (eds.) Dynamic, Genetic, and Chaotic Programming, pp. 203–321. John Wiley, New York, 1992.
- [23]C.G. LANGTON, Artificial Life: Proceedings of an Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems. Addison-Wesley Longman Publishing Co., Inc, Boston, MA, USA, 1989.
- [24]C.G. LANGTON, Proceedings of the Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems (ALIFE '87), Los Alamos, NM, USA, September 1987, Volume 6 of Santa Fe Institute Studies in the Sciences of Complexity, Addison- Wesley. (pp. 15, 18), 1989.
- [25]F. LAMARCHE, S. DONIKIAN, Automatic Orchestration of Behaviours through the management of Resources and Priority Levels, in Autonomous Agents and Multi Agent Systems, Bologna, Italy, July 2002.
- [26]N. LASSABE, H. LUGA, Y. DUTHEN, Evolving creatures in virtual ecosystems., Pan, Z., Cheok, A.D., Haller, M., Lau, R.W.H., Saito, H., Liang, R. (eds.) ICAT 2006. LNCS, vol. 4282, pp. 11–20. Springer, Heidelberg, 2006.
- [27]N. LASSABE, H. LUGA, Y. DUTHEN, A New Step for Evolving Creatures, IEEE Symposium on Artificial Life (IEEE-ALife 2007), Honolulu, Hawaii, IEEE, p. 243-251, april 2007.
- [28]O. LE ROUX, Modélisation déclarative d'environnements virtuels : contribution à l'étude des techniques de génération par contraintes. PhD thesis, Université Paul Sabatier, 2003.
- [29]H. LIPSON, Curious and creative machines., "50 Years of AI", pp. 316– 320. (p. 2), 2007.
- [30]H. LUGA, Y. DUTHEN, R. PELLE, A. BERRO, Extended algebraic surfaces generation for volume modeling: an approach through genetic algorithms, Proceedings of Visualization and Modeling, Leeds, British Computer Society, 1995.

- [31]D. PANZOLI, H. LUGA, Y. DUTHEN, Introducing an Associative Memory in a Neural Controller for Advanced Situated Agents, International Conference on Computer Graphics and Artificial Intelligence (3IA 2007), Athens. Greece, Dimitri Plemenos (Eds.), Laboratoire XLIM - Université de Limoges, p. 137-149, may 2007.
- [32]T.S. RAY, Aesthetically evolved virtual pets. In Artificial Life 7 workshop proceedings, pp. p158–161. (pp. 27, 33, 36), 2000.
- [33]K. SIMS, Artificial evolution for computer graphics, SIGGRAPH '91, Proceedings of the 18th annual conference on Computer graphics and interactive techniques, pp. 319–328. ACM Press, New York, NY, USA 1991.
- [34]K. SIMS, Evolving 3d morphology and behavior by competition, Artificial Life IV Proceedings, pp. 28–39. MIT Press, Cambridge, 1994.
- [35]S. SANCHEZ, H. LUGA, Y. DUTHEN, Learning classifier systems and behavioural animation of virtual characters., Gratch, J., Young, M., Aylett, R., Ballin, D., Olivier, P. (eds.) IVA 2006. LNCS (LNAI), vol. 4133, p. 467. Springer, Heidelberg, 2006.
- [36]S. SANCHEZ, O. ROUX, H. LUGA, H., V. GAILDRAT, Constraint-Based 3D Object Layout using a Genetic Algorithm, 3IA'2003, The Sixth International Conference on Computer Graphics and Artificial Intelligence, Limoges, 14/05/03-15/05/03, 2003.
- [37]D. TERZOPOULOS,X. TU, Artificial fishes : Autonomous locomotion, perception, behavior, and learning in a simulated physical world. Artificial Life 1, 4, 327–351. (p. 19), 1994.
- [38]S. TODD, W. LATHAM Evolutionary Art and Computers, Academic Press, Inc, Orlando, FL, USA, 1994.
- [39]A. TURING, The chemical basis of morphogenesis, Philosophical Transactions of the Royal Society of London, Series B, No.641, Vol. 237, 14 August 1952.
- [40]J. VENTRELLA, Explorations in the emergence of morphology and locomotion behavior in animated characters. In Proceedings of the 4th International Workshop on the Synthesis and Simulation of Living Systems Artificial Life IV (Cambridge, MA, USA), pp. 436–441. MIT Press. (p. 47), 1994.
- [41]J. VENTRELLA, Designing emergence in animated artificial life worlds, VW '98: Proceedings of the first International Conference on Virtual Worlds, London, UK, pp. 143–155. Springer-Verlag. (pp. 17, 47), 1998.
- [42]M.WHITELAW, Metacreation: Art and Artificial Life. MIT Press, Cambridge, MA, USA, 2004.
- [43]S.W. WILSON, ZCS: a zeroth level classifier system. Evolutionary Computation, 2(1), 1-18 (1994).Untel, A. *Titre du livre*. L'Armada, Paris, 2005.